Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 249 (2009) 307–331
www.elsevier.com/locate/entcs

Twisted Graph States for Ancilla-driven Universal Quantum Computation

E. Kashefi[a], D. K. L. Oi[b], D. Browne,[c]
J. Anders[c] and E. Andersson[d]

[a] *School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, UK*
[b] *SUPA, Department of Physics, University of Strathclyde, Glasgow G4 0NG, UK*
[c] *Department of Physics & Astronomy, University College London, London WC1E 6BT, UK*
[d] *SUPA, Department of Physics, Heriot-Watt University, Edinburgh EH14 4AS, UK*

Abstract

We introduce a new paradigm for quantum computing called Ancilla-Driven Quantum Computation (ADQC) which combines aspects both of the quantum circuit [1] and the one-way model [2] to overcome challenging issues in building large-scale quantum computers. Instead of directly manipulating each qubit to perform universal quantum logic gates or measurements, ADQC uses a fixed two-qubit interaction to couple the memory register of a quantum computer to an ancilla qubit. By measuring the ancilla, the measurement-induced back-action on the system performs the desired logical operations.
The underlying mathematical model is based on a new entanglement resource called *twisted graph states* generated from non-commuting operators, leading to a surprisingly powerful structure for parallel computation compared to graph states obtained from commuting generators. [3]. The ADQC model is formalised in an algebraic framework similar to the Measurement Calculus [4]. Furthermore, we present the notion of *causal flow* for twisted graph states, based on the stabiliser formalism, to characterise the determinism. Finally we demonstrate compositional embedding between ADQC and both the one-way and circuit models which will allow us to transfer recently developed theory and toolkits of measurement-based quantum computing and quantum circuit models directly into ADQC.

Keywords: Quantum computation, ancilla-driven, universal quantum computation, graph states

1 Introduction

There are two main paradigms which have driven both the theory and implementation of quantum computation; gate-based quantum computing (GBQC) [1], and measurement-based quantum computing (MBQC) [2]. Though these two models are computationally equivalent, in practice each has their own advantages and disadvantages which have major implications for the choice of physical system, design, and operation. We introduce a new paradigm called ancilla-driven quantum computing which combines features of both mentioned models, in order to parallelise the architecture of quantum computers, to decrease decoherence effects, and simplify their physical implementation and operation.

doi:10.1016/j.entcs.2009.07.096

GBQC requires, in general, arbitrary dynamic global operations which in turn complicates the design and characterisation of the entire computer. Additionally it would be desirable to physically separate preparation, measurement, and coherent interaction regions to reduce control complexity, circuitry congestion, and decoherence due to cross-talk. In contrast, MBQC performs computation purely through single-qubit measurement on a pre-existing static multi-partite entangled state, distilling and processing non-local correlation. However, the generation of the initial highly entangled state, incorporation of quantum error correction and fault-tolerance, and individual measurement of each qubit are issues in many candidate systems.

Our new model is partly inspired by the previous works of Andersson and Oi in [5], and also Perdrix and Jorrand in [6]. The first paper introduced an efficient method to implement any generalised quantum measurement by coupling the system with an ancilla qubit. However the method remains essentially similar to GBQC as it assumes arbitrary dynamic coupling operations between ancilla and system. On the other hand, the second paper describes a probabilistic version of MBQC in terms of a Turing machine where one can view the read-write head playing the role of an ancilla qubit, though this is not the way that the paper presents the model. Moreover this approach still directly manipulates the memory register and requires dynamic global measurement operators.

Ancilla-driven quantum computing (ADQC) attempts to overcome such issues by performing computation where the memory register (input data) can only be remotely manipulated through interaction with a supply of prepared ancillas. In other words, instead of directly manipulating data qubits to perform universal quantum logic gates or measurements, ADQC uses a fixed two-qubit unitary interaction to couple the memory register of a quantum computer to an ancilla qubit. By measuring the ancilla, the measurement-induced back-action on the system performs the desired logical operation. Practically, a single fixed unitary interaction coupling the data and ancilla qubits greatly simplifies design, construction, and operation of the computer since only one particular discrete operation needs to be generated and characterised. Furthermore, separating interaction and measurement leads to a parallel structure with possibly reduced decoherence. A requisite interaction for universal ADQC already exists in a variety of physical systems ranging from ion micro-traps, neutral atoms, nuclear spin donors in semiconductors, SQUIDs and cavity QED which greatly increases the scope for implementation of the core ideas. ADQC also naturally benefits from optimisation of the qubit species employed for memory and ancilla. Memory qubits can be chosen for long coherence time at the expense of being static and difficult to manipulate directly, whilst ancilla qubits may be chosen for high mobility and rapid initialisation and measurement, e.g. donor nuclear spins in isotopically pure silicon as memory and electron spins conveyed via charge transport by adiabatic passage (CTAP) as ancilla in solid state quantum computing.

So far we have discussed only the practical advantages of our proposed architecture. The formalisation of the computational model underlying ADQC, which is the

focus of the current paper, leads to the introduction of a new multi-partite entanglement resource. Only recently has it been demonstrated that a very restricted class of multi-partite entangled states are useful for universal deterministic MBQC [7,8]. However a full characterisation of such states [9] remains an open problem which this paper aims to make progress upon.

The entangled graph states [3] have emerged as an elegant and powerful quantum resource, especially for measurement-based quantum computation (MBQC) [2]. Many important results on their entanglement properties [10], information flow [11,12], implementation [13], and novel applications in cryptography [14,15], are due to their deceptively simple description. The generating operator for graph states, called controlled-phase, is a symmetric and commuting operator which leads to a simple graphical notation and hence the name for these states. Additionally, the elegant result by van de Nest *et al.* [16,10] shows that any stabiliser state is equivalent to a graph state up to local Clifford operators. This greatly expands the scope of these results and leads to a natural extension of the above constructions into stabiliser states, as well as allowing a convenient graphical notation for a very general class of states. If we consider *open* graph states, graph states where some nodes (called input nodes) are given in arbitrary states (rather than being prepared in a particular fixed state which is the case for graph states) much of the theory still follows. However open stabiliser states with arbitrary input nodes no longer fulfil the pre-requisites of the theorem by van de Nest *et. al.*, and in general they do not admit a trivial graphical notation.

We address in this paper a particular class of open stabiliser states, called *twisted graph states* which, despite having a non-commuting generator, still admits a simple graph representation. They form the key ingredient for ADQC. We then show how this new class of states can be viewed as open graph states up to some local swap operations. We also develop an algebraic framework similar to the measurement calculus, which is the mathematical framework underlying MBQC computation, to derive the standardisation theory for the ADQC patterns of computation. As we will see, any ADQC computation requires a classical control structure to compensate for the probabilistic nature of the measurement. We introduce the notion of *causal flow* for twisted graph states based on the stabiliser formalism, to characterise the determinism. Compared to the open graph state, the stabiliser state has a more complicated and global structure. One can however sometimes construct computation within ADQC that is more parallel than other existing quantum models. We will demonstrate this fact with a simple example. The full study of the parallel power of the model is, however, outside the scope of this paper. Finally we construct direct translations between ADQC and MBQC for a subclass of deterministic patterns with flow. We also present the embedding between GBQC and ADQC and show how a separation in depth can be obtained.

We have presented the required preliminaries on quantum computing and necessary concepts from both gate-based and measurement-based models in the appendix to make the paper accessible for a general audience.

2 Ancilla-Driven Model

As motivated in the introduction in ancilla-driven quantum computing we are interested in the following two essential properties:

- The only global operation is a fixed interaction between ancilla and system.
- Only ancilla qubits will be measured.

We introduce the ADQC model within an algebraic framework similar to that of the measurement calculus recalled in the appendix. We have a set of fixed basic commands described below where the indices $i, j, \ldots$ represent the qubits on which each of these operations apply. A *pattern* is a sequence of commands defined over a set of qubits in the list V, called *computation space*, where the particular sub-list S represents the *system* qubits (we may refer to them as data or memory register) and the rest $A = V \setminus S$ are the *ancilla* qubits. In what follows we define an arbitrary pure single qubit state by

$$|+_{\theta,\phi}\rangle = \cos(\tfrac{\theta}{2})|0\rangle + e^{i\phi}\sin(\tfrac{\theta}{2})|1\rangle\,,$$

and denote its orthogonal state (the opposite point in the Bloch Sphere) with

$$|-_{\theta,\phi}\rangle = \sin(\tfrac{\theta}{2})|0\rangle - e^{i\phi}\cos(\tfrac{\theta}{2})|1\rangle\,,$$

where $0 \leq \theta \leq \pi$ and $0 \leq \phi \leq 2\pi$.

- **Preparation.** $N_a^{|\psi\rangle}$ $(a \in A)$ prepares an ancilla qubit in the state $|\psi\rangle$.
- **Interaction.** $\widetilde{E}_{as}$ $(s \in S, a \in A)$ entangle a system qubit and an ancilla qubit with interaction operator controlled-Z followed by Hadamard on each qubit: $\widetilde{\wedge Z} := H_s \otimes H_a \wedge Z_{as}$ [1].
- **Ancilla Measurement.** $M_a^{\lambda,\alpha}$ $(a \in A)$ measures qubit a on plane $\lambda \in \{(X,Y),(X,Z),(Y,Z)\}$, defined by orthogonal projections into:
  · $|\pm_{(X,Y),\alpha}\rangle := |\pm_{\frac{\pi}{2},\alpha}\rangle$ if $\lambda = (X,Y)$
  · $|\pm_{(X,Z),\alpha}\rangle := |\pm_{\alpha,0}\rangle$ if $\lambda = (X,Z)$
  · $|\pm_{(Y,Z),\alpha}\rangle := |\pm_{\alpha,\frac{\pi}{2}}\rangle$ if $\lambda = (Y,Z)$
  with the convention that $|+_{\theta,\phi}\rangle\langle+_{\theta,\phi}|_a$ corresponds to the outcome $m_a = 0$, while $|-_{\theta,\phi}\rangle\langle-_{\theta,\phi}|_a$ corresponds to $m_a = 1$. The propagation of dependent corrections (next command) defines dependent measurement:

  $$_n[M_a^{\lambda,\alpha}]^m := M_a^{\lambda,\alpha} X_a^m Z_a^n$$

  where $m, n, \ldots$ are module 2 summation of several measurements outcomes, also called *signals*. The *domain* of a signal is the set of qubits on which it depends. [2]

[1] We can also consider the interaction of the form controlled-Z + SWAP with the same initial ancilla and measurements, similar results can be derived. The choice of interaction depends on the natural dynamics of the physical implementation

[2] Depending on the context we sometimes use the notation m for syntax, i.e., a set of qubits (representing a formal sum) and sometimes for semantics, i.e., 0 or 1.

- **Corrections.** X_i and Z_i ($i \in V$), 1-qubit Pauli operators. As in MBQC, to control the non-determinism of the measurement outcomes certain local corrections will depend upon previous measurement outcomes. These will be written as C_i^m, with $C_i^0 = I$, and $C_i^1 = C_i$.

We write $\mathfrak{H}_V$ for the associated quantum state space $\otimes_{i\in V}\mathbb{C}^2$. To run a pattern, one prepares the system qubits in some given input state $\Psi \in \mathfrak{H}_S$, while the ancilla qubits are all prepared according to the N commands in fixed $|\psi\rangle$ states. The commands are then executed in sequence, and finally the result of the pattern computation is read back from the system qubits[3]. Clearly, for this procedure to succeed, we had to impose the definiteness conditions as stated in the appendix.

The main differences between ADQC and MBQC are (1) The interaction operator being $\widetilde{\wedge Z}$ instead of $\wedge Z$, which still belongs to the normaliser of the Pauli group; (2) Only ancilla qubits can be measured, that is to say in the terminology of MBQC any ADQC pattern has the same number of inputs and outputs which are overlapping (system qubits). Apart from universality, which we will prove later, most of the theory of measurement calculus [4] which was developed for the one-way quantum computer can be easily adapted to ADQC. For completeness we briefly review this here.

The first way to combine patterns is by composing them. Two patterns $\mathfrak{P}_1$ and $\mathfrak{P}_2$ may be composed if $S_1 = S_2$. Provided that $\mathfrak{P}_1$ has as many system qubits as $\mathfrak{P}_2$, by renaming these qubits, one can always make them composable. However it is important to emphasise that since the $\widetilde{E}_{ij}$ operators are non-commuting their order of appearance in each pattern must be preserved under the renaming and composition. The other way of combining patterns is to tensor them. Two patterns $\mathfrak{P}_1$ and $\mathfrak{P}_2$ may be tensored if $V_1 \cap V_2 = \emptyset$. Again one can always meet this condition by renaming qubits in a way that these sets are made disjoint.

2.1 The semantics of patterns

We present a formal operational semantics for ADQC patterns as a probabilistic labelled transition system, similar to [4]. Besides quantum states, one needs a classical state recording the outcomes of the successive measurements one does in a pattern. If we let U stand for the finite set of qubits that are still active (i.e. not yet measured) and W stands for the set of qubits that have been measured (i.e. they are now just classical bits recording the measurement outcomes), it is natural to define the computation state space as:

$$\mathcal{C} := \Sigma_{U,W}\mathfrak{H}_U \times \mathbb{Z}_2^W.$$

In other words the computation states form a U, W-indexed family of pairs q, Γ, where q is a quantum state from $\mathfrak{H}_U$ and Γ is a map from some W to the outcome space $\mathbb{Z}_2$. We call this classical component Γ an *outcome map*, and denote by $\emptyset$ the

[3] Preparation and readout of the system qubits can be performed by using suitable ancilla states and measurements.

empty outcome map in $\mathbb{Z}_2^{\emptyset}$. We need further preliminary notation. For any signal m and classical state $\Gamma \in \mathbb{Z}_2^W$, such that the domain of m is included in W, we take m_Γ to be the value of m given by the outcome map Γ. That is to say, if $m = \sum_I m_i$, then $m_\Gamma := \sum_I \Gamma(i)$ where the sum is taken in $\mathbb{Z}_2$. Also if $\Gamma \in \mathbb{Z}_2^W$, and $x \in \mathbb{Z}_2$, we define

$$\Gamma[x/i](i) = x,\ \Gamma[x/i](j) = \Gamma(j) \text{ for } j \neq i$$

which is a map in $\mathbb{Z}_2^{W \cup \{i\}}$.

We may now view each of our commands as acting on the state space $\mathcal{C}$:

$$\begin{array}{lll}
q, \Gamma & \xrightarrow{N_i^{|\psi\rangle}} & q \otimes |\psi\rangle_i, \Gamma \\
q, \Gamma & \xrightarrow{\widetilde{E}_{ij}} & \widetilde{\wedge Z}_{ij} q, \Gamma \\
q, \Gamma & \xrightarrow{X_i^m} & X_i^{m_\Gamma} q, \Gamma \\
q, \Gamma & \xrightarrow{Z_i^m} & Z_i^{m_\Gamma} q, \Gamma \\
U \cup \{i\}, W, q, \Gamma & \xrightarrow{{}^n[M_i^{\lambda,\alpha}]^m} & U, W \cup \{i\}, \langle +_{\lambda,\alpha_\Gamma}|_i q, \Gamma[0/i] \\
U \cup \{i\}, W, q, \Gamma & \xrightarrow{{}^n[M_i^{\lambda,\alpha}]^m} & U, W \cup \{i\}, \langle -_{\lambda,\alpha_\Gamma}|_i q, \Gamma[1/i]
\end{array}$$

where $\alpha_\Gamma = (-1)^{m_\Gamma}\alpha + n_\Gamma \pi$. We introduce an additional command called *signal shifting*:

$$q, \Gamma \xrightarrow{F_i^{m_\Gamma}} q, \Gamma[\Gamma(i) + m_\Gamma / i]$$

It consists in shifting the measurement outcome at i by the amount m_Γ. Note that the Z-action leaves measurements globally invariant, in the sense that $|+_{\alpha+\pi}\rangle, |-_{\alpha+\pi}\rangle = |-_\alpha\rangle, |+_\alpha\rangle$. Thus changing α to $\alpha + \pi$ amounts to replacing the outcomes of the measurements, and one has:

$$ {}_{n_\Gamma}[M_i^\alpha]^{m_\Gamma} = F_i^{n_\Gamma}\, {}_0[M_i^\alpha]^{m_\Gamma} \tag{1}$$

and signal shifting allows us to dispose of the Z action of a measurement, sometimes resulting in convenient optimisations of standard forms. In the rest of the paper, for simplicity, we omit the superscript Γ on the measurement outcomes.

The usual convention has it that when one does a measurement the resulting state is *renormalised* and the probabilities are associated with the transition. We do not adhere to this convention here, instead we leave the states unnormalized. The reason for this choice of convention is that this way, the probability of reaching a given state can be read off its norm, and the overall treatment is simpler.

2.2 Denotational Semantics

We now present the denotational semantics of ADQC patterns. If n is the number of measurements then the run may follow 2^n different branches. Each branch is associated with a unique binary string $\mathbf{n}$ of length n, representing the classical outcomes of the measurements along that branch, and a unique *branch map* $A_{\mathbf{n}}$ representing the linear transformation from $\mathfrak{H}_S$ to $\mathfrak{H}_S$ along that branch. This map

is obtained from the (un-normalised) operational semantics via the sequence (q_i, Γ_i) with $1 \leq i \leq m$ (where m is the total number of commands), such that:

$$q_1, \Gamma_1 = q \otimes |+ \ldots +\rangle, \emptyset$$
$$\text{and for all } i \leq m : q_{i-1}, \Gamma_{i-1} \xrightarrow{A_i} q_i, \Gamma_i.$$

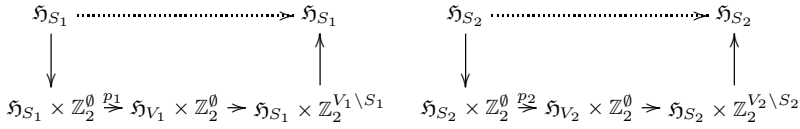
and all measurement commands in the sequence $\{A_i\}$ have been replaced by appropriate projections corresponding to the outcome index $\mathbf{n}$.

Definition 2.1 A pattern $\mathfrak{P}$ realizes a map on density matrices ρ given by $\rho \mapsto \sum_{\mathbf{s}} A_{\mathbf{s}}(\rho) A_{\mathbf{s}}^{\dagger}$. We write $[\![\mathfrak{P}]\!]$ for the map realised by $\mathfrak{P}$.

It is then easy to prove [4] that each pattern realizes a completely positive trace preserving (CPTP) map and if a pattern is strongly deterministic (see appendix), then it realizes a unitary embedding [4]. Hence the denotational semantics of a pattern is a CPTP-map. It is also compositional, as the following theorem shows.

Theorem 2.2 *For two patterns* $\mathfrak{P}_1$ *and* $\mathfrak{P}_2$ *we have* $[\![\mathfrak{P}_1\mathfrak{P}_2]\!] = [\![\mathfrak{P}_2]\!][\![\mathfrak{P}_1]\!]$ *and* $[\![\mathfrak{P}_1 \otimes \mathfrak{P}_2]\!] = [\![\mathfrak{P}_2]\!] \otimes [\![\mathfrak{P}_1]\!]$.

Proof. Recall that two patterns $\mathfrak{P}_1$, $\mathfrak{P}_2$ may be combined by composition provided $\mathfrak{P}_1$ has as many system qubits as $\mathfrak{P}_2$. Suppose this is the case, and suppose further that $\mathfrak{P}_1$ and $\mathfrak{P}_2$ respectively realise some CPTP-maps T_1 and T_2. We need to show that the composite pattern $\mathfrak{P}_2\mathfrak{P}_1$ realizes T_2T_1. Indeed, the two diagrams representing branches in $\mathfrak{P}_1$ and $\mathfrak{P}_2$:



can be pasted together, since $S_1 = S_2$, and $\mathfrak{H}_{S_1} = \mathfrak{H}_{S_2}$. But then, it is enough to notice 1) that preparation steps p_2 in $\mathfrak{P}_2$ commute with all actions in $\mathfrak{P}_1$ since they apply on disjoint sets of qubits, and 2) that no action taken in $\mathfrak{P}_2$ depends on the measurements outcomes in $\mathfrak{P}_1$. It follows that the pasted diagram describes the same branches as does the one associated to the composite $\mathfrak{P}_2\mathfrak{P}_1$. A similar argument applies to the case of a tensor combination, and one has that $\mathfrak{P}_2 \otimes \mathfrak{P}_1$ realizes $T_2 \otimes T_1$. □

2.3 Generating patterns

In order to prove the universality we present two simple generic patterns where only (X, Y) plane measurement (M^{α}), Pauli Z measurement (M^{Z}) and $|+\rangle$ ancilla preparation N are sufficient. Note that a Pauli Z measurement can be considered as a special case of an (X, Z) or (Y, Z) plane measurement with $\alpha = 0$.

The following one-parameter family $J(\alpha)$ generates all single-qubit unitary operators [17]:

$$J(\alpha) := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & e^{i\alpha} \\ 1 & -e^{i\alpha} \end{pmatrix}$$

as any unitary operator U on $\mathbb{C}^2$ can be written:

$$U = e^{i\alpha} J(0)J(\beta)J(\gamma)J(\delta)$$

for some α, β, γ and δ in $\mathbb{R}$. Recall that the MBQC implementation of the J generator is:

$$\mathfrak{J}(-\alpha) := X_a^{m_1} M_s^{(X,Y),\alpha} E_{sa} \tag{2}$$

where s is the system qubit input and a is the ancilla and E_{sa} is controlled-Z operator [17]. On the other hand we can write the following pattern which also implements the J gate but now the ancilla qubit a will be measured

$$\mathfrak{J}(-\alpha) := H_s Z_s^{m_a} M_a^{(Y,Z),\alpha} E_{sa} \tag{3}$$

where H_s represents the application of a Hadamard gate on the system qubit. We can now manipulate the new pattern to derive a generating pattern for J operator in our model. Hence we have to get rid of the H_s and replace the interaction command with $\widetilde{E}_{sa}$.

$$\begin{aligned} \mathfrak{J}(-\alpha) &:= H_s Z_s^{m_a} M_a^{(Y,Z),\alpha} E_{sa} \\ &= X_s^{m_a} M_a^{(Y,Z),\alpha} H_s E_{sa} \\ &= X_s^{m_a} M_a^{(X,Y),\alpha} H_a H_s E_{sa} \\ &= X_s^{m_a} M_a^{(X,Y),\alpha} \widetilde{E}_{sa} \end{aligned} \tag{4}$$

Now we only need a generator for a two-qubit unitary such as controlled-Z to obtain the full universality but the MBQC pattern for controlled-Z_{ij} (E_{ij} with $i, j \in S$) is not desirable as it is an operator between two qubits of the system rather than an interaction between system and ancilla qubits. Therefore the natural choice instead is to consider interaction of type $\widetilde{E}_{as'}\widetilde{E}_{as}$ and it is easy to check that the Pauli Z measurement of the ancilla will give us a simple generating pattern for the two qubit operator $\widetilde{\wedge Z}_{ss'}$:

$$\widetilde{\wedge\mathfrak{Z}} := X_s^{m_a} M_a^Z \widetilde{E}_{as'} \widetilde{E}_{as} \tag{5}$$

Any unitary can then be simulated by sequential and parallel compositions of the above generating patterns, where the composition simply glues given patterns over the common system qubits while preserving the initial orders of the commands. We will return to the important issue of how to represent the composed pattern graphically but to do so first we have to address the important feature of the ADQC model that is the standardisation procedure which permits us to rewrite any well defined patterns, e.g. obtained from composition, to be put in the standard form where all the preparation commands are applied first followed by the entangling, measurement, and finally correction commands.

For simplicity, in the remainder of this paper we will restrict ourselves to a special class of patterns, namely, those using only ancillas of degree 1 with arbitrary (X, Y) plane measurement and of degree 2 with Pauli Z measurement. However, the whole theory developed in this paper can be easily extended to the more general setting.

2.4 Standardisation

It is known that any MBQC model can admit a standardisation procedure if and only if the entangling command belongs to the normaliser group of the group generated by the correction commands [12]. This is the case for our ADQC model and the following are the required rewrite rules:

$$\widetilde{E}_{ij}X_i^s = X_j^s Z_i^s \widetilde{E}_{ij} \tag{6}$$

$$\widetilde{E}_{ij}Z_i^s = X_i^s \widetilde{E}_{ij} \tag{7}$$

The rules for propagation of the correction through measurement are the same as for MBQC with additional rules for the M^Z measurement:

$${}_n[M_a^\alpha]^m X_a^p = {}_n[M_a^\alpha]^{m+p} \tag{8}$$

$${}_n[M_a^\alpha]^m Z_a^p = {}_{n+p}[M_a^\alpha]^m \tag{9}$$

$$M_a^Z X_a^m = F_a^m M_a^Z \tag{10}$$

$$M_a^Z Z_a^m = M_a^Z \tag{11}$$

We also have the same free commutation rewrite rules:

$$\widetilde{E}_{ij}A_{\boldsymbol{k}} \Rightarrow A_{\boldsymbol{k}}\widetilde{E}_{ij} \quad \text{where } A \text{ is not an entanglement} \tag{12}$$

$$A_{\boldsymbol{k}}X_i^m \Rightarrow X_i^m A_{\boldsymbol{k}} \quad \text{where } A \text{ is not a correction} \tag{13}$$

$$A_{\boldsymbol{k}}Z_i^m \Rightarrow Z_i^m A_{\boldsymbol{k}} \quad \text{where } A \text{ is not a correction} \tag{14}$$

where $\boldsymbol{k}$ represent the qubits acted upon by command A, and are distinct from i and j.

Recall that the effect of a Z correction on a qubit a simply flips the outcome of a measurement to be made on that qubit. Hence we can replace the dependencies induced by the Z correction by appropriate operations over the measurement outcomes as described below. In what follows $m[n/m_i]$ denotes the substitution of m_i with n in m where m, n are modulo 2 summations of several measurement outcomes,

$${}_n[M_a^\alpha]^m = F_a^n[M_a^\alpha]^m \tag{15}$$

$$X_j^m F_i^n = F_i^n X_j^{m[n+m_i/m_i]} \tag{16}$$

$$Z_j^m F_i^n = F_i^n Z_j^{m[n+m_i/m_i]} \tag{17}$$

$${}_n[M_j^\alpha]^m F_i^p = F_i^p \, {}_{n[p+m_i/m_i]}[M_j^\alpha]^{m[p+m_i/m_i]} \tag{18}$$

$$F_i^m F_j^n = F_j^n F_i^{m[n+m_j/m_j]}. \tag{19}$$

One can then use the exact same method as in the case of MBQC to prove that this rewrite system has the desired properties of confluence and termination.

It is important to emphasise the main difference between ancilla-driven and MBQC which is the interaction command $\widetilde{E}_{ij}$ versus E_{ij}. In order to achieve the desirable features of not directly measuring system qubits and having a fixed interaction, we had to give up the simple operator E_{ij} which is the generating operator of the open graph state. [4] As we show in the appendix, for any standard pattern in MBQC we can write its underlying open graph state with qubits representing the nodes and E_{ij} the edges of the graph. Then remarkably only from the geometry of this graph we can obtain the dependency structures to guarantee a deterministic computation in MBQC. In other words the simple graph representation for the global operation defining pattern allows one to determine dynamic properties directly from the static structure. Can we still obtain similar properties for our new model? Despite the non-commutativity of $\widetilde{E}_{ij}$ the answer is yes. In the next section we define the twisted graph state which is the underlying geometry of a given ancilla-driven pattern obtained from standardisation and we present how one can directly construct the dependency structure from their geometry.

3 Twisted Graph States

The main issue with the $\widetilde{E}_{ij}$ operators is the fact that they are non-commuting, therefore after standardisation their order will be important. Another important property of an ancilla-driven pattern is that system qubits interact only with ancilla qubits. Therefore we introduce a multipartite entangled state as a graph over ancilla and system qubits and $\widetilde{E}_{ij}$ edges with extra condition to address the above mentioned requirements. In Section 4.2 we show how this new class of states can be viewed as open graph states up to some local swap operators. This is the reason behind the chosen name.

Definition 3.1 An *open twisted graph state* $(G, S, A, \mathcal{C})$ consists of a bipartite graph G over disjoint sets of qubits S and A, called *systems* and *ancillas*, such that the maximum degree of any ancilla node is 2, together with an edge labelling $\mathcal{C}$. The labels define a partial ordering over edges where the order is total and strict on any edges that share a common vertex, *i.e.* it defines an edge colouring.

The corresponding quantum state, denoted as $|\widetilde{E}_G\rangle$, is obtained by preparing qubits in S in given arbitrary states and all the qubits in A in the $|+\rangle$ state, and then applying $\widetilde{E}_{ij}$ over corresponding qubits according to the partial order of $\mathcal{C}$ (see Figure 1).

One may think of an open twisted graph state as the beginning of the definition of an ancilla-driven pattern, where one has already decided how many qubits will be used ($V = S \cup A$) and how they will be entangled:

$$\widetilde{E}_G := \prod_{\{i,j\}\in\tilde{E}} \widetilde{E}_{ij}.$$

[4] Informally speaking this is due to the fact that using E_{ij} and ancilla measurement one cannot implement a Hadamard gate on the system qubits. However the full characterisation of the operators that lead to universal ADQC is outside the scope of the current paper and will appear in a forthcoming publication.

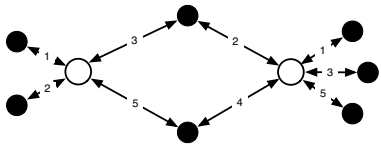


Fig. 1. An open twisted graph state where the system qubits are the white circles that will not be measured and the rest are ancilla qubits. The edges are $\widetilde{E}$ interactions and edge labels denote the partial order.

To complete the definition of the pattern it remains to decide which angles will be used to measure ancilla qubits and, most importantly if one is interested in determinism, which dependent corrections will be applied. Conversely, any ancilla-driven pattern has a unique underlying open twisted graph state obtained by forgetting measurements and corrections where the colour of the edges is given by the most partial order of the $\widetilde{E}_{ij}$ commands which respects the non-commuting order. Recall that two $\widetilde{E}_{ij}$ commands commute if and only if they act on disjoint set of qubits. Note that the depth of this partial order is the true depth of the preparation of the state.

Different ordered colourings over the same graph structure might lead to different twisted graph states and consequently different patterns of computation. We leave as an open question whether one can find a more relaxed definition that can still uniquely define the entangled state corresponding to an ancilla-driven pattern. On the other hand our restrained definition will allow us to derive the dependency structure of the measurements directly from the order of the colouring, this is the topic of the next subsection.

3.1 Dependency structure

We will use the graph stabiliser formalism [18,3] to construct a deterministic pattern. Recall that for any open graph state $|E_G\rangle$ defined over a graph G with vertices V, we have the following set of equations for all the non-input qubits:

$$X_i \prod_{j\in G(i)} Z_j(|E_G\rangle) = |E_G\rangle$$

where $G(i)$ is the set of neighbour vertices of i in G. The above Pauli operators are called the stabiliser operators of $|\psi\rangle$.

Similarly we define the stabiliser operators of a given twisted graph state $|\widetilde{E}_G\rangle$ defined over a graph G with vertices $V = S \cup A$. We will use the following notation as well. Define $S(a)$ for $a \in A$ to be the attached system qubit $s \in S$ with the smallest edge label and $S'(a)$ to be the other one if it exists; $N(s)$ for $s \in S$ to be the set of ancilla qubits connected to the system qubit s; and finally $G_{S(a)}$ to be the sub-graph with edges between $S(a)$ and $N_{S(a)}$.

Consider first a simple case where $\widetilde{E}_G = \widetilde{E}_{aS(a)} N_a^{|+\rangle}$. Then the stabiliser has the form

$$Z_a X_{S(a)}(|\widetilde{E}_G\rangle) = |\widetilde{E}_G\rangle \tag{20}$$

The above equation is due to

$$\begin{aligned} Z_a X_{S(a)}(\widetilde{E}_{aS(a)} N_a^{|+\rangle}) &= \widetilde{E}_{aS(a)} X_a Z_{S(a)} Z_{S(a)} N_a^{|+\rangle} \\ &= \widetilde{E}_{aS(a)} N_a^{|+\rangle} \end{aligned}$$

and for another simple case of $|\widetilde{E}_G\rangle = \widetilde{E}_{aS'(a)}\widetilde{E}_{aS(a)} N_a^{|+\rangle}$ we have

$$X_a X_{S(a)}(|\widetilde{E}_G\rangle) = |\widetilde{E}_G\rangle \tag{21}$$

again due to

$$\begin{aligned} X_a X_{S(a)}(\widetilde{E}_{aS'(a)}\widetilde{E}_{aS(a)} N_a^{|+\rangle}) &= \widetilde{E}_{aS'(a)}\widetilde{E}_{aS(a)} X_a Z_{S(a)} Z_{S(a)} N_a^{|+\rangle} \\ &= \widetilde{E}_{aS'(a)}\widetilde{E}_{aS(a)} N_a^{|+\rangle}. \end{aligned}$$

In order to generalise the above cases the following rewrite rules for Pauli commutations will be used:

- Equation 6, $\widetilde{E}_{ij} X_i^s = X_j^s Z_i^s \widetilde{E}_{ij}$, transforms the X operation on the system qubit to the next immediate ancilla qubit, introducing a Z operation at the system qubit.
- Equation 7, $\widetilde{E}_{ij} Z_i^s = X_i^s \widetilde{E}_{ij}$, replaces the Z operation on the system qubit with an X operation.

Unlike the stabiliser of the graph state which has a local structure, in the case of a twisted graph state the stabiliser of a affects the whole of the graph. This action is, however, recursive and can be defined using the collection of several local actions. Define the label of an ancilla node to be the same as the label of the edge $\widetilde{E}_{aS(a)}$. Consider an ancilla qubit a and those qubits in $N(S(a))$ with label greater than label of a (see Figure 2). We can assume, without loss of generality, that all edges connected to $S(a)$ have labels 1 to n with 1 being the label of a. This is due to the fact that the stabiliser of a has an effect only on those qubits in $N(S(a))$ where their edge interaction are after the edge interaction of a and $S(a)$ hence having a greater edge label.

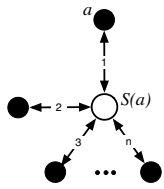


Fig. 2. The generic case for the local study of stabiliser at ancilla a.

Definition 3.2 Given a twisted graph state $(G, S, A, \mathcal{C})$, a *local stabiliser* on the ancilla qubit a is defined as following:

- Consider vertices in $G_{S(a)}$ with labels greater than label of a and relabel them from 1 to n according to $\mathcal{C}$ ordering, with 1 being the label of a.

- Add X on vertices in $N(S(a))$ with even label and degree 1.
- Add Z on vertices in $N(S(a))$ with even label and degree 2.
- Add X on the system qubit if n is odd otherwise add Z.

We denote the above set of Pauli operators with $P_l(S(a))$ which acts on a subset of qubits in $N(S(a))$.

Define $I(a)$ to be the set of degree-two ancilla qubits in $N(S(a))$ with even label that get a Z Pauli operator in the definition of the local stabiliser of a. The stabiliser of a will have the same local effect as defined above over $S(a')$ for all the $a' \in I(a)$ and the same structure repeats for vertices in $I(a')$. Therefore we define recursively such qubits

$$I^*(a) = \{a' | \exists n : a' \in I^n(a)\}$$

where $I^1(a) = I(a)$ and $I^{n+1}(a) = \bigcup_{a' \in I^n(a)} I(a')$. We can now present a recursive definition for the stabiliser of a twisted graph state as the product of a collection of local stabilisers.

Definition 3.3 Given a twisted graph state $(G, S, A, \mathcal{C})$, the *stabiliser* on the ancilla qubit a is defined as follows:

$$P(a) = Z_a \prod_{a' \in I^*(a)} P_l(S(a')) \quad \text{if } a \text{ has degree one}$$
$$P(a) = X_a \prod_{a' \in I^*(a)} P_l(S(a')) \quad \text{otherwise}$$

It is a straightforward but cumbersome computation to show the correctness of the above definition and we omit the details of the proof.

We can now adapt the notion of flow and generalised flow of graph states [11,12] for twisted graph states to derive a sufficient condition for determinism. The key idea is exactly the same as in the MBQC case based on the following simple observation. We could make a measurement $M_i^{(X,Y),\alpha}$ "deterministic" (corrected) if it could be pre-composed by an anachronical $Z_i^{s_i}$ correction (*i.e.* conditioned on the outcome of a measurement which hasn't happened yet). This unphysical scenario is a useful starting point for our proof.

$$\langle +_{(X,Y),\alpha}|_a = M_a^{(X,Y),\alpha} Z_a^{m_a}$$

The flow construction guarantees that a deterministic pattern with anachronical corrections

$$\mathfrak{P} = \prod_{a \in A}^{\mathcal{C}} \ \langle +_{(X,Y),\alpha}|_a \ \widetilde{E}_G$$

$$= \prod_{a \in A}^{\mathcal{C}} \ M_a^{(X,Y),\alpha} Z_a^{m_a} \ \widetilde{E}_G$$

can be transformed into a runnable pattern, where all dependencies will respect the proper causal ordering. The key observation which allows us to transform this into a runnable pattern is that the flow construction defines a stabiliser $P_{f(a)}$ which when

composed with the anachronical correction, forms an operator which commutes with the measurement, and thus the pattern can be brought into runnable order.

For simplicity in the rest of the paper we consider only patterns where degree-two vertices are measured with Pauli Z and degree-one vertices are measured in the (X,Y) plane, we use the generic term $M_a^{\lambda_a,\alpha_a}$ for both cases. This class of patterns are large enough to introduce a universal ADQC model as they include the generating pattern introduced in Section 2.3. However the definition of flow and determinism can be extended to the more general case as well.

Definition 3.4 An open twisted graph state $(G,S,A,\mathcal{C})$ has *causal flow* if there exists a partial order $>$ over V consistent with the ordering of $\mathcal{C}$ such that for all $a \in A$ and all vertices $a' \in P(a)$ we have $a < a'$ except for those a' that will be measured with Pauli Z.

Theorem 3.5 *Suppose the open twisted graph state $(G,S,A,\mathcal{C})$ has a causal flow with the partial order $>$. Define:*

$$C(a) = P(a)Z_a \quad \textit{for all degree-one ancilla } a$$
$$C(a) = P(a)X_a \quad \textit{for all degree-two ancilla } a$$

then the pattern:

$$\mathfrak{P}_{G,\boldsymbol{\alpha}} := \prod_{a\in A}^{>} \; C(a)^{m_a} \; M_a^{\lambda_a,\alpha_a} \; \widetilde{E}_G$$

where the product follows the dependency order $>$, is runnable, uniformly and strongly deterministic.

Proof. The proof is based on the following equations first consider the (X,Y) measurement case for degree-one ancillas:

$$\begin{aligned}\langle +_\alpha|_a(\widetilde{E}_G) &= M_a^\alpha Z_a^{m_a}(\widetilde{E}_G)\\ &= M_a^\alpha Z_a^{m_a} P(a)^{m_a}(\widetilde{E}_G)\\ &= C(a)^{m_a} M_a^\alpha(\widetilde{E}_G)\end{aligned}$$

Similarly for the Z measurement over degree-two ancillas we have:

$$\begin{aligned}\langle 0|_a(\widetilde{E}_G) &= M_a^Z X_a^{m_a}(\widetilde{E}_G)\\ &= M_a^Z X_a^{m_a} P(a)^{m_a}(\widetilde{E}_G)\\ &= C(a)^{m_a} M_a^Z(\widetilde{E}_G)\end{aligned}$$

Hence we can write:

$$\prod_{a\in A}^{>}\langle \lambda_a,\alpha_a|_a \widetilde{E}_G = \prod_{a\in A}^{>} \; C(a)^{m_a} \; M_a^{\lambda_a,\alpha_a} \; \widetilde{E}_G.$$

The left hand side is clearly a uniformly and strongly deterministic pattern. The right hand side pattern is runnable as the introduced corrections follow the partial

order $>$ except for the Z correction introduced over degree-two ancillas. However one can ignore them since these qubits will be measured with Pauli Z and we have $M_a^Z\ Z_a^m = M_a^Z$. This finishes the proof. □

It is interesting to note that the flow definition for a graph state was based on the geometry of the underlying graph, whereas in a twisted graph state it is based on the edge colouring order. Indeed, as mentioned before, different edge colourings lead to different twisted graph states and hence different flow constructions. Roughly speaking, the edge colouring plays the role of geometry for the twisted graph states.

4 Compositional Embedding

One of the main foci in constructing direct translations between models is to study parallelism as the more parallel the computation, the more robust it is against decoherence. Recently the advantage of MBQC over GBQC in terms of depth complexity has been demonstrated where a logarithmic separation was shown [19]. Furthermore it is also known that the parallel power of MBQC is equivalent to GBQC equipped with quantum fan-out gates [20]. Such a full analysis for ADQC is outside the scope of this paper. We will only present the transformations between ADQC and other models which could be used for pattern synthesis and present only an example on how ADQC could be more parallel.

4.1 GBQC and ADQC

The question of translating GBQC circuits into MBQC patterns and vice versa has been addressed before in [19] and it can be directly adapted for ADQC as well. In fact the universality proof of ADQC already presents a method of translation of a circuit into ADQC: (I) Rewrite the given circuit in terms of the universal gates set of $J(\alpha)$ and $\widetilde{\wedge Z}$; (II) Replace each gate with its corresponding ADQC pattern (equations 4 and 5); (III) Perform the standardisation procedure.

The above construction cannot be used in reverse as the edge colouring order might lead to a circuit with an acausal loop. However it is possible to keep all the auxiliary qubits to avoid creating loops in the resulting circuit. The scheme is simply based on the well-known method of coherently implementing a measurement [19]. It is also easy to prove, in a similar way as in [19], that the translation from a GBQC circuit into an ADQC pattern will never increase the depth as the number of the edge colouring number of the obtained twisted graph state will be upper-bounded by the depth of the original circuit. More importantly, we present an example where the depth decreases exponentially. Consider the ladder structure of the circuit in Figure 3 which has depth n. This circuit, through the introduced construction, will be translated into a pattern with the twisted graph state shown in Figure 3 which has constant depth 4. This is due to the simple fact that both the computation and preparation depths for any ADQC pattern are upper bounded by the edge colouring of the graph, which in this case is equal to 4.

Note that the same depth separation result between GBQC and MBQC obtained for the parity function [19] is also valid for the case of GBQC and ADQC. How-

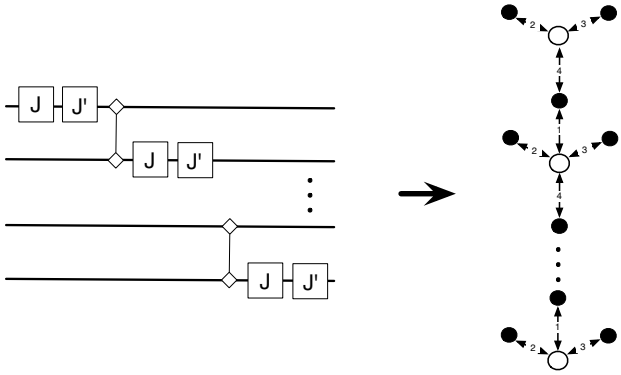


Fig. 3. A ladder structure circuit with the binary $\widetilde{\wedge Z}$ gates and the unitary $J(\alpha)$ gates, together with the corresponding open twisted graph state obtained via gate by gate translation.

ever the nature of the above example differs from the class of circuits that can be parallelised via a translation into MBQC. In fact a direct translation of the above example will lead to a pattern of depth n (see Figure 4). Hence the example suggests that we might achieve more parallelism in the ADQC model in comparison to GBQC and MBQC and a detailed study of depth complexity in ADQC might reveal further new techniques in parallelising GBQC as it was done for MBQC [19].

4.2 MBQC and ADQC

As mentioned before there exists a compositional embedding from MBQC patterns with flow into GBQC and vice versa, and together with the construction of the last subsection one can obtain an embedding between ADQC and MBQC for patterns with flow. However in view of parallelism, it is interesting to find such an embedding directly by presenting the correspondence between twisted open graph states and open graph states.

The following equation relates the two resources but it is only valid for degree-one ancilla qubits

$$\widetilde{E}_{as} N_a^{|+\rangle} = \mathrm{SWAP}_{a,s}\, E_{as} N_a^{|+\rangle} \tag{22}$$

where $\mathrm{SWAP}_{a,s}$ is the unitary operator swapping qubits a and s. This equation and the next one are in fact the reason behind the chosen name for this class of states as one can recover a graph state from them by applying the appropriate sequence of twist operators. In order to handle the degree-two ancilla qubits we will use the following pattern equations

$$\begin{aligned}
\widetilde{\wedge\mathfrak{Z}} &:= X_s^{m_a} M_a^Z \widetilde{E}_{as'} \widetilde{E}_{as} \\
&= X_s^{m_a} M_a^Z H_a H_{s'} E_{as'} \widetilde{E}_{as} \\
&= X_s^{m_a} M_a^X H_{s'} E_{as'} \widetilde{E}_{as} \\
&= X_{s'}^{m_b} X_s^{m_a} M_a^X M_b^X \widetilde{E}_{bs'} E_{as'} \widetilde{E}_{as}
\end{aligned} \tag{23}$$

In the new pattern for $\widetilde{\wedge\mathfrak{Z}}$ both instances of $\widetilde{E}_{as}$ can be replaced using Equation 22. Therefore we can replace any pattern over a given twisted graph state where

degree-two vertices are measured with Pauli Z, into a pattern over a graph state obtained through the above manipulations of the $\widetilde{E}_{as}$ edges.

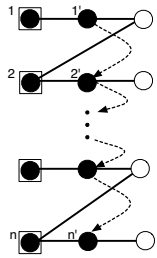


Fig. 4. The MBQC pattern obtained from the direct translation of the circuit in Figure 3. The pattern has depth n due to the sequence of dependent measurements (the doted lines).

The other direction of translation between patterns, from MBQC into ADQC, is an open problem. One can of course translate any MBQC pattern with flow first into a GBQC circuit and then to a corresponding ADQC but we believe a direct embedding, if it can be found, can reveal more insights on parallelism and also the relationship between commuting graph states and non-commuting twisted graph states as resources. We finish this section by pointing out again that an ADQC pattern might be indeed more parallel than the corresponding MBQC one. Recall the ladder circuit in Figure 3, where the corresponding ADQC pattern had constant depth 4. Now as shown in Figure 4 the depth of the MBQC pattern obtained from a direct translation will be n due to the sequence of the X dependencies between measurements on qubits $1', 2', \cdots, n'$. This begs the question of direct translation between MBQC and ADQC and a careful analysis of depth trade off. We leave as a conjecture that one might find a logarithmic depth separation between these two models.

5 Discussion

5.1 Physical Implementations

ADQC presents significant advantages over GBQC for particular physical implementations. By isolating the system memory from measurement and state preparation, the physical layout of a quantum computer can be optimised. Potentially decoherent read-out mechanisms can be located away from the memory. Since only a fixed two-qubit unitary gate has to be implemented between memory and ancilla qubits, this simplifies considerably the construction, characterisation, control and operation of the computer. Control line clutter can be reduced, due to relaxing the need to implement single qubit rotations and measurement on the memory, simplifying the architecture of the computer as well as minimising the possibility of cross-talk. The choice of physical qubits for memory and ancilla can also be optimised. Memory qubits can be chosen for long coherence times at the expense of being static, whilst ancilla qubit can be chosen for mobility, and ease of initialisation and measurement.

Natural candidate systems consist of an array of static qubits addressed by flying

qubits or a "read/write" head. For example, an optical lattice of neutral atoms can be addressed by a separately controlled atom [21,22]. Since it is difficult to individually address with lasers a single site of a fully filled optical lattice, the read-write head would interact with selected sites and can be used as the ancilla. A similar idea can be applied to neutral atoms trapped in dipole trap arrays [23] controlled by an atom in an optical tweezer [24]. Similarly in [25], ions in an array of micro traps would be manipulated by a single ion read-write head. Alternatively in [26], the role of the memory is played by the quantised electromagnetic field in an array of cavities whilst Rydberg atoms traversing the cavities act as ancillas (similar ideas are contained in [27]). All of the above schemes possess a natural control-Z (dispersive) interaction between the ancilla and system qubits. For universality, either an additional Hadamard operation on the system should be incorporated into this interaction (e.g. through the pushing laser in [25]), or an effective SWAP operation be found in conjunction with the Controlled-Z. The latter could be achieved through cold collisions between ancilla and system [28,29].

A system which may prove particularly amenable to our model is one based on [30]. Here, the nuclear spin of a single dopant atom in isotopically pure silicon plays the role of a memory qubit which can be controllably coupled via the hyperfine interaction to an electron spin which acts as an ancilla qubit. Nuclear spins can be very well isolated from the environment, as well as the state preparation and measurement areas. Electrons can be rapidly transported around the computer using charge transport via adiabatic passage (CTAP), this avoids the issue of swapping nuclear spin states, as in the original Kane proposal [31], which can lead to a reduction in fault tolerance. An issue here is making sure that the interaction between electron and nuclear spins is of the correct form as to allow conditional unitary dynamics[5]. A controlled-Z + SWAP gate can be achieved by using the method presented in [32] with only the Heisenberg interaction between ancilla and system and local operations on the ancilla itself.

A natural interaction which is also suitable for ADQC is the XY-Hamiltonian which be easily turned into the controlled-Z + SWAP gate [33] (equivalent to the ISWAP in the aforementioned reference). In [34], a natural XY-interaction between nuclear spins, as in the Kane proposal, is mediated by a 2D electron gas in the Quantum Hall regime. The XY-interaction also naturally occurs between quantum dots coupled by a cavity [35] or superconducting qubits [36,37]. Here, the memory and ancilla qubits are of the same species.

5.2 Open Problems

The model for ADQC presented here has been based upon either $\widetilde{\wedge Z} := H_s \otimes H_a \wedge Z_{as}$ or control-Z + SWAP. These interactions have the useful property of permitting unitary conditional measurement-induced back-action. However, this property is shared by a more general class of interactions. It is an open problem as to

[5] In order for the conditional measurement-induced back-action on the system to be unitary, the non-local party of the unitary interaction between system and ancilla must be of a certain form. A more thorough examination of which unitary interactions are sufficient or necessary for universal ADQC will be the subject of another paper.

which interactions lead to universal ADQC. We may even look at generalising our computational models to dispense with determinism (whether different branches of the computation can be corrected by simple Pauli operators) or even unitarity at each stage.

More generally, it is interesting to expand the set of resource states universal for quantum computation [7,8]. Different interactions may lead to different classes of states, together with their own measurement calculus. We can envisage connecting the resource states for universal MBQC with physical interactions which generate them, and hence the type of correlations which are induced. From the ADQC embedding, we can trace the roles of measurement-induced back-action and the non-local character of the generating interaction.

6 Conclusion

Ancilla-Driven Quantum Computation presents a new way of performing universal quantum computing. Aside from potential advantages for quantum computer construction and operation, it leads to a new set of universal quantum computational resources, the twisted graph state, based upon a non-commuting generating interaction. The measurement calculus has been developed to encompass this new model and even greater parallelism compared to conventional MBQC may be possible. Despite the non-commuting nature of the generating interaction for twisted-graph states, the graphical structure still encodes the dynamics (dependancy structure) of the computation. A strong possibility is that ADQC could further improve the parallelism of MBQC. By further studying ADQC and its generalisations, further insight into this question could be achieved.

References

[1] D. Deutsch. Quantum computational networks. *Proc. Roy. Soc. Lond A*, 425, 1989.

[2] R. Raussendorf and H. J. Briegel. A one-way quantum computer. *Physical Review Letters*, 86, 2001.

[3] M. Hein, J. Eisert, and H. J. Briegel. Multi-party entanglement in graph states. *Physical Review A*, 69, 2004. quant-ph/0307130.

[4] V. Danos, E. Kashefi, and P. Panangaden. The measurement calculus. *Journal of ACM*, 2007.

[5] E. Andersson and D. K. L. Oi. Binary search trees for generalized measurements. *Physical Review A*, 77(5, Part A), 2008.

[6] S. Perdrix and P. Jorrand. Measurement-based quantum Turing machines and their universality. quant-ph/0404146, 2004.

[7] D. Gross, S. Flammia, and J. Eisert. Most quantum states are too entangled to be useful as computational resources. arXiv:0810.4331v2, 2008.

[8] M. J. Bremner, C. Mora, and A. Winter. Are random pure states useful for quantum computation? arXiv:0812.3001v1, 2008.

[9] D. Gross, J. Eisert, N. Schuch, and D. Perez-Garcia. Measurement-based quantum computation beyond the one-way model. *Physical Review A*, 76, 2007.

[10] M. Van den Nest, W. Duer, A. Miyake, and H. J. Briegel. Fundamentals of universality in one-way quantum computation. *New Journal of Physics*, 9, 2007.

[11] V. Danos and E. Kashefi. Determinism in the one-way model. *Physical Review A*, 2006.

[12] D. Browne, E. Kashefi, M. Mhalla, and S. Perdrix. Generalized flow and determinism in measurement-based quantum computation. *New Journal of Physics*, 9, 2007.

[13] S. D. Barrett and P. Kok. Efficient high-fidelity quantum computation using matter qubits and linear optics. *Physical Review A*, 71, 2005.

[14] D. Markham and B. C. Sanders. Graph states for quantum secret sharing. *Physical Review A*, 78(4), 2008.

[15] A. Broadbent, J. Fitzsimons, and E. Kashefi. Universal blind quantum computation. arXiv:0807.4154v1, 2008.

[16] M. Van den Nest, J. Dehaene, and B. De Moor. An efficient algorithm to recognize local clifford equivalence of graph states. *Physical Review A*, 70, 2004. quant-ph/0405023.

[17] V. Danos, E. Kashefi, and P. Panangaden. Parsimonious and robust realizations of unitary maps in the one-way model. *Physical Review A*, 72, 2005.

[18] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[19] A. Broadbent and E. Kashefi. On parallelizing quantum circuits. quant-ph/0704.1736, 2007.

[20] D. Browne and S. Perdrix. On the computational power the measurement-based quantum computing. Private communication, 2009.

[21] L You and MS Chapman. Quantum entanglement using trapped atomic spins. *Physical Review A*, 62(5):art. no.–052302, 2000.

[22] T Calarco, U Dorner, PS Julienne, CJ Williams, and P Zoller. Quantum computations with atoms in optical lattices: Marker qubits and molecular interactions. *Physical Review A*, 70(1), 2004.

[23] R Dumke, M Volk, T Muther, F.B.J. Buchkremer, G Birkl, and W Ertmer. Micro-optical realization of arrays of selectively addressable dipole traps: A scalable configuration for quantum computation with atomic qubits. *Physical Review Letters*, 89(9), 2002.

[24] J. Beugnon, C. Tuchendler, H. Marion, A. Gaetan, Y. Miroshnychenko, Y. R. P. Sortais, A. M. Lance, M. P. A. Jones, G. Messin, A. Browaeys, and P. Grangier. Two-dimensional transport and transfer of a single atomic qubit in optical tweezers. *Nature Physics*, 3(10):696–699, 2007.

[25] J.L. Cirac and P. Zoller. A scalable quantum computer with ions in an array of microtraps. *Nature*, 404(6778):579–581, 2000.

[26] V Giovannetti, D Vitali, P Tombesi, and A Ekert. Scalable quantum computation with cavity QED systems. *Physical Review A*, 62(3), 2000.

[27] P. J. Blythe and B. T. H. Varcoe. A cavity-QED scheme for cluster-state quantum computing using crossed atomic beams. *New Journal of Physics*, 8, 2006.

[28] E Charron, E Tiesinga, F Mies, and C Williams. Optimizing a phase gate using quantum interference. *Physical Review Letters*, 88(7), 2002.

[29] K Eckert, J Mompart, XX Yi, J Schliemann, D Bruss, G Birkl, and M Lewenstein. Quantum computing in optical microtraps based on the motional states of neutral atoms. *Physical Review A*, 66(4), 2002.

[30] AD Greentree, JH Cole, AR Hamilton, and LCL Hollenberg. Coherent electronic transfer in quantum dot systems using adiabatic passage. *Physical Review B*, 70(23), 2004.

[31] BE Kane. A silicon-based nuclear spin quantum computer. *Nature*, 393(6681):133–137, 1998.

[32] D Loss and DP DiVincenzo. Quantum computation with quantum dots. *Physical Review A*, 57(1):120–126, 1998.

[33] N Schuch and J Siewert. Natural two-qubit gate for quantum computation using the XY interaction. *Physical Review A*, 67(3), 2003.

[34] D Mozyrsky, V Privman, and ML Glasser. Indirect interaction of solid-state qubits via two-dimensional electron gas. *Physical Review Letters*, 86(22):5112–5115, 2001.

[35] A Imamoglu, DD Awschalom, G Burkard, DP DiVincenzo, D Loss, M Sherwin, and A Small. Quantum information processing using quantum dot spins and cavity QED. *Physical Review Letters*, 83(20):4204–4207, 1999.

[36] J Siewert, R Fazio, GM Palma, and E Sciacca. Aspects of qubit dynamics in the presence of leakage. *J. Low Temperature Physics*, 118(5-6):795–804, 2000. International Conference on Electron Transport in Mesoscopic Systems (ETSM '99), GOTHENBURG, SWEDEN, AUG 12-15, 1999.

[37] L.S. Levitov, T.P. Orlando, J.B. Majer, and J.E. Mooij. Quantum spin chains and majorana states in arrays of coupled qubits. arXiv:cond-mat/0108266v2, 2001.

[38] R.P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21, 1982.

[39] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. In *Proceedings of the Royal Society of London*, volume A400, 1985.

[40] R. Raussendorf and H. J. Briegel. Computational model underlying the one-way quantum computer. *Quantum Information & Computation*, 2, 2002. quant-ph/0108067.

[41] R. Raussendorf, D. E. Browne, and H. J. Briegel. Measurement-based quantum computation on cluster states. *Physical Review A*, 68, 2003.

[42] Richard Jozsa. An introduction to measurement based quantum computation. quant-ph/0508124, 2005.

[43] M. A. Nielsen. Cluster-state quantum computation. *Reviews in Mathematical Physics*, 2005. quant-ph/0504097.

[44] D. E. Browne and H. J. Briegel. One-way quantum computation - a tutorial introduction. quant-ph/0603226, 2006.

A Preliminaries

We briefly review the required concepts from quantum computing, a more detailed introduction can be found in [18]. Let $\mathcal{H}$ denote a 2-dimensional complex vector space, equipped with the standard inner product. We pick an orthonormal basis for this space, label the two basis vectors $|0\rangle$ and $|1\rangle$, and for simplicity identify them with the vectors $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$, respectively. A *qubit* is a unit length vector in this space, and so can be expressed as a linear combination of the basis states:

$$\alpha_0|0\rangle + \alpha_1|1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}.$$

Here α_0, α_1 are complex *amplitudes*, and $|\alpha_0|^2 + |\alpha_1|^2 = 1$.

An *m-qubit state* is a unit vector in the m-fold tensor space $\mathcal{H} \otimes \cdots \otimes \mathcal{H}$. The 2^m basis states of this space are the m-fold tensor products of the states $|0\rangle$ and $|1\rangle$. We abbreviate $|1\rangle \otimes |0\rangle$ to $|1\rangle|0\rangle$ or $|10\rangle$. With these basis states, an m-qubit state $|\phi\rangle$ is a 2^m-dimensional complex unit vector

$$|\phi\rangle = \sum_{i \in \{0,1\}^m} \alpha_i |i\rangle.$$

There exist quantum states that cannot be written as the tensor product of other quantum states, such states are called *entangled* states, *e.g.* $|00\rangle + |11\rangle$.

We use $\langle\phi| = |\phi\rangle^*$ to denote the conjugate transpose of the vector $|\phi\rangle$, and $\langle\phi \mid \psi\rangle = \langle\phi| \cdot |\psi\rangle$ for the inner product between states $|\phi\rangle$ and $|\psi\rangle$. These two states are *orthogonal* if $\langle\phi \mid \psi\rangle = 0$. The *norm* of $|\phi\rangle$ is $|| \phi ||= \sqrt{\langle\phi \mid \phi\rangle}$.

A quantum state can evolve by a unitary operation or by a measurement. A *unitary* transformation is a linear mapping that preserves the norm of the states. If we apply a unitary U to a state $|\phi\rangle$, it evolves to $U|\phi\rangle$. The *Pauli operators* are a well-known set of unitary transformations for quantum computing:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

and the *Pauli group* on n qubits is generated by Pauli operators. Several other unitary transformations that we will use in this paper are: the identity I, the *phase* gate $P(\alpha)$, of which $P(\pi/4)$ and $P(\pi/2)$ are a special cases, the Hadamard H and the controlled-Z ($\wedge Z$):

$$I := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad P(\alpha) := \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \quad H := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\wedge Z := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad SWAP := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The *Clifford group* on n qubits is generated by the matrices Z, H, $P(\pi/2)$ and $\wedge Z$, and is the normaliser of the Pauli group. This set of matrices is not universal for quantum computation, but by adding any single-qubit gate not in the Clifford group (such as $P(\pi/4)$), we do get a set that is approximately universal for quantum computing.

The most general measurement allowed by quantum mechanics is specified by a family of positive semi-definite operators $E_i = M_i^* M_i$, $1 \leq i \leq k$, subject to the condition that $\sum_i E_i = I$. A projective measurement is defined in the special case where the operators E_i are projections. Let $|\phi\rangle$ be an m-qubit state and $\mathcal{B} = \{|b_1\rangle, \ldots, |b_{2^m}\rangle\}$ an orthonormal basis of the m-qubit space. A projective measurement of the state $|\phi\rangle$ in the $\mathcal{B}$ basis means that we apply the projection operators $P_i = |b_i\rangle\langle b_i|$ to $|\phi\rangle$. The resulting quantum state is $|b_i\rangle$ with probability $p_i = |\langle\phi \mid b_i\rangle|^2$. An important class of projective measurements are Pauli measurements, *i.e.* projections to eigenstates of Pauli operators.

So far we have dealt with *pure* quantum states. A more general representation with density matrices also allows us to describe open physical systems, where one can prepare a classical stochastic mixture of pure quantum states, called *mixed* quantum states. For a system in a pure state $|\psi\rangle$, the density matrix is just the

projection operator $|\psi\rangle\langle\psi|$. Suppose that we only know that a system is one of several possible states $|\psi_1\rangle, \ldots, |\psi_k\rangle$ with probabilities $p_1, \ldots, p_k$ respectively. We define the density matrix for such a state to be

$$\rho = \sum_{i=1}^{k} p_i |\psi_i\rangle\langle\psi_i|.$$

The most general physical operator that acts over density matrices is a completely positive trace preserving map (CPTP) $\mathcal{E} : \mathcal{B}(\mathfrak{H}_1) \to \mathcal{B}(\mathfrak{H}_2)$ with Kraus decomposition

$$\mathcal{E}(\rho) = \sum_m A_m \rho A_m^\dagger$$

where the $A_m : \mathfrak{H}_1 \to \mathfrak{H}_2$, $\mathcal{B}(\mathfrak{H})$ is the Banach space of bounded linear operators and we require that

$$\sum_m A_m^\dagger A_m = I$$

A.1 Quantum circuit model

Richard Feynman was one of the first to suggest that a computer based on the principles of quantum mechanics could efficiently *simulate* other quantum systems [38]. David Deutsch then developed the idea that the quantum computer could offer a computational advantage compared to a classical computer; he also defined the *quantum Turing machine* [39], before defining the *quantum circuit model* [1] to represent quantum computations.

Any unitary operation U can be approximated with a circuit C, using gates from a fixed universal set of gates. The *size* of a circuit is the number of gates and its *depth* is the largest number of gates on any input-output path. Equivalently, the depth is the number of layers that are required for the parallel execution of the circuit, where a qubit can be involved in at most one interaction per layer. In this paper, we adopt the model according to which at any given time-step, a single qubit can be involved in at most one interaction. This differs from the *concurrency* viewpoint, according to which all interactions for commuting operations can be done simultaneously.

A.2 Measurement-based model

We give a brief introduction to measurement-based quantum computing (MBQC) [2,40,41], a more detailed description is available in [42,43,44,4] and our notation follows that of [4]. In MBQC, computations are represented as *patterns*, which are sequences of *commands* acting on the qubits in the pattern. These commands are of four types:

(i) N_i is a one-qubit preparation command which prepares the auxiliary qubit i in state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. The preparation commands can be implicit from the pattern: when not specified, all non-input qubits are prepared in the $|+\rangle$ state.

(ii) E_{ij} is a two-qubit entanglement command which applies the controlled-Z operation, $\wedge Z$, to qubits i and j. Note that the $\wedge Z$ operation is symmetric and so $E_{ij} = E_{ji}$. Also, E_{ij} commutes with E_{jk} and so the ordering of the entanglement commands in not important.

(iii) M_i^α is a one-qubit measurement on qubit i which depends on parameter $\alpha \in [0, 2\pi)$ called the *angle of measurement*. M_i^α is the orthogonal projection onto states

$$|+_\alpha\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\alpha}|1\rangle)$$
$$|-_\alpha\rangle = \frac{1}{\sqrt{2}}(|0\rangle - e^{i\alpha}|1\rangle),$$

followed by a trace-out operator, since measurements are destructive. We denote the classical outcome of a measurement performed at qubit i by $m_i \in \mathbb{Z}_2$. We take the specific convention that $m_i = 0$ if the measurement outcome is $|+_\alpha\rangle$, and that $m_i = 1$ if the measurement outcome is $|-_\alpha\rangle$. Outcomes can be summed together resulting in expressions of the form

$$m = \sum_{i \in I} m_i$$

which are called *signals*, and where the summation is understood as being done modulo 2. The *domain* of a signal is the set of qubits on which it depends (in this example, the domain of m is I).

(iv) X_i and Z_i are one-qubit Pauli corrections which correspond to the application of the Pauli X and Z matrices, respectively, on qubit i.

In order to obtain universality, we have to add a classical control mechanism called *feed-forward*, which allows measurement angles and corrections to be dependent on the results of previous measurements [2,4]. Let m and n be signals. Dependent corrections are written as X_i^m and Z_i^n and dependent measurements are written as ${}_n[M_i^\alpha]^m$. The meaning of dependencies for corrections is straightforward: $X_i^0 = Z_i^0 = I$ (no correction is applied), while $X_i^1 = X_i$ and $Z_i^1 = Z_i$. In the case of dependent measurements, the measurement angle depends on m, n and α as follows:

$${}_n[M_i^\alpha]^m = M_i^{(-1)^m\alpha + n\pi} \tag{A.1}$$

so that, depending on the parity of m and n, one may have to modify the angle of measurement α to one of $-\alpha$, $\alpha + \pi$ and $-\alpha + \pi$. These modifications correspond to conjugations of measurements under X and Z:

$$X_i^m M_i^\alpha X_i^m = M_i^{(-1)^m\alpha} \tag{A.2}$$
$$Z_i^n M_i^\alpha Z_i^n = M_i^{\alpha + n\pi} \tag{A.3}$$

and so we will refer to them as the X- and Z-actions or alternatively as the X- and Z-dependencies. Since measurements are destructive, the above equations simplify

to:

$$M_i^\alpha X_i^m = M_i^{(-1)^m \alpha} \tag{A.4}$$
$$M_i^\alpha Z_i^n = M_i^{\alpha + n\pi}. \tag{A.5}$$

Note that these two actions are commuting, since $-\alpha + \pi = -\alpha - \pi$ up to 2π, and hence the order in which one applies them does not matter.

A *pattern* is defined by the choice of a finite set V of qubits, two not necessarily disjoint sets $I \subseteq V$ and $O \subseteq V$ determining the pattern inputs and outputs, and a finite sequence of commands acting on V. We require that no command depend on an outcome not yet measured, that no command act on a qubit already measured, that a qubit be measured if and only if it is not an output qubit and that a qubit be prepared if and only if it is not an input qubit. This set of rules is known as the *definiteness* condition.

A pattern is said to be in *standard form* if all the preparation commands N_i and entanglement operators E_{ij} appear first in its command sequence, followed by measurements and finally corrections. A pattern that is not in standard form is called a *wild pattern*. Any wild pattern can be put in its unique standard form [4]; this form can reveal implicit parallelism in the computation. The procedure of rewriting a pattern in its standard form is called *standardisation*. This can be done by applying the following rewrite rules:

$$E_{ij} X_i^m \Rightarrow X_i^m Z_j^m E_{ij} \tag{A.6}$$
$$E_{ij} Z_i^m \Rightarrow Z_i^m E_{ij} \tag{A.7}$$
$${}_n[M_i^\alpha]^m X_i^p \Rightarrow {}_n[M_i^\alpha]^{m+p} \tag{A.8}$$
$${}_n[M_i^\alpha]^m Z_i^p \Rightarrow {}_{n+p}[M_i^\alpha]^m \tag{A.9}$$

The rewrite rules also contain the following *free commutation rules* which tell us that, if we are dealing with disjoint sets of target qubits, measurement, corrections and entanglement commands commute pairwise [4].

$$E_{ij} A_{\boldsymbol{k}} \Rightarrow A_{\boldsymbol{k}} E_{ij} \quad \text{where } A \text{ is not an entanglement} \tag{A.10}$$
$$A_{\boldsymbol{k}} X_i^m \Rightarrow X_i^m A_{\boldsymbol{k}} \quad \text{where } A \text{ is not a correction} \tag{A.11}$$
$$A_{\boldsymbol{k}} Z_i^m \Rightarrow Z_i^m A_{\boldsymbol{k}} \quad \text{where } A \text{ is not a correction} \tag{A.12}$$

where $\boldsymbol{k}$ represent the qubits acted upon by command A, and are distinct from i and j. Standardisation allow us to present graphically the global operation of a pattern. We define an *open graph state* (G, I, O) to consist of an undirected graph G together with two subsets of nodes I and O, called inputs and outputs. We write V for the set of vertices in G, E for the set of edges, I^c, and O^c for the complements of I and O in V and $E_G := \prod_{\{i,j\} \in E} E_{ij}$ for the global entanglement operator associated with G. Trivially, any standard pattern has a unique underlying open graph state, obtained by forgetting measurements and correction commands.